

Problem 1. π

The following is the Leibniz formula for π :

$$\sum_0^{\infty} \frac{(-1)^n}{2n+1} = \frac{\pi}{4}.$$

Write a program (using a `for`-loop) to compute

$$4 \sum_0^{1000} \frac{(-1)^n}{2n+1},$$

and then print an approximation for π using the result of the computation.

See https://en.wikipedia.org/wiki/Leibniz_formula_for_%CF%80 for details on why this works.

Hint: the following code computes

$$\sum_0^{500} n = 0 + 1 + 2 + 3 + \dots + 500 :$$

```
res = 0
for n in range(501): # the last number to be added in should be 500
    res = res + n
```

For this problem, you are doing the same thing, but for a more complicated summand.

Problem 2. A bit more π

Now, write a program using a `while`-loop to compute the quantity from Problem 1. See Monday's lecture here <https://youtu.be/HwkrQupE7LE?t=1898> for a technique for converting `for`-loops into `while`-loops. ""

Problem 3.

Review the solutions for Lab #2. If you haven't successfully implemented the `undo` functionality, implement it now. Look at the solutions as much as you need to, and seek explanations from the TAs as necessary, but make sure to get to a point where you definitely can do `undo` from scratch.

Problem 4.

Add the function `undo2` to your implementation. The function `undo2` will change the current value to the value before last. For example:

```
    # current_value is 0
+1   # current_value is 1
*2   # current_value is 2
undo # current_value is 1
+ 5  # current_value is 6
undo2 # current_value is 2
```

Hint: when implementing `undo`, we kept track of `prev_value`. Now, you need to keep track of an additional variable.

Problem 5.

Implement a simple simulation that's similar in spirit to Project 1. This is a simulation of an EngSci student. The only functions are:

```
def drink_coffee():
    # your code here

def study(minutes):
    # your code here
```

The following initialize function is given:

```
def initialize():
    global too_much_coffee
    global current_time
    global last_coffee_time
    global last_coffee_time2
    global knols
    too_much_coffee = False
    current_time = 0
    knols = 0
    last_coffee_time = -100
    last_coffee_time2 = -100
```

The rules are: the student gets 5 knols per minute of study (“knols” are units of knowledge) if they haven't drunk coffee right before the study session, and 10 knols per minute of study if they had a coffee right before the study session.

However, if at any time the student drinks more than 2 cups of coffee over the span of 2 hours, they get 0 knols per hour from there on.

Here is how the code could be used:

```
if __name__ == '__main__':
    initialize()      # start the simulation
    study(60)        # knols = 300
    study(20)        # knols = 400
    drink_coffee()   # knols = 400
    study(10)        # knols = 500
    drink_coffee()   # knols = 500
    study(10)        # knols = 600
    drink_coffee()   # knols = 600, 3rd coffee in 20 minutes
    study(10)        # knols = 600
```

To approach this problem, think of the following:

- Every time the student studies, you need an if-statement to determine how to update `knols`
- When drinking coffee, need to update the coffee times
- Every time you're studying, need to update the current time

Problem 6.

Work on Project 1.